

In the Claims

1. (Previously Presented) A method of converting data of a plurality of input data types to a plurality of output data types by an application program, said method comprising:
 - (a) receiving a first attribute of a first input data type and a second attribute of a first output data type;
 - (b) dynamically creating at runtime a first optimized conversion routine based on said first attribute and said second attribute, the conversion routine including one or more computer instructions to be executed during conversion;
 - (c) validating specific field conversion options of the conversion routine; and
 - (d) executing said first optimized conversion routine from said application program to convert data of said first input data type to said first output data type.
2. (Previously Presented) The method of claim 1, wherein step (d) comprises calling said first optimized conversion routine from said application.
3. (Previously Presented) The method of claim 1, wherein step (d) comprises storing said first optimized conversion routine inline with said application.
4. (Original) The method of claim 1, wherein step (b) is performed dynamically while said application program is executing.
5. (Previously Presented) The method of claim 1, further comprising:
 - (e) receiving a third attribute of a second input data type and a fourth attribute of a second output data type;
 - (f) generating a second optimized conversion routine based on said third attribute and said fourth attribute; and
 - (g) executing said second optimized conversion routine from said application program to convert input data of said second input data type to said second output data type.
6. (Original) The method of claim 1, wherein said first and second attribute is character type.

7. (Original) The method of claim 1, further comprising generating program debugging instrumentation for said first optimized conversion routine.

8. (Currently amended) A method of converting data from input data types to output data types, said method comprising:

(a) receiving a plurality of sets of input attributes and output attributes from an application program, each set comprising one or more input attributes and one or more output attributes, the input attributes in a set being associated with a first data type and the output attributes in the set being associated with a second data type different than the first data type;

(b) dynamically creating at runtime a ~~plurality of~~ data conversion routines for each set of input attributes and output attributes, ~~the data conversion routines for converting data from one data type having an input attribute to another data type having an output attribute, the one data type having an input attribute not being dependent on the other data types having an output attribute, the~~ each conversion routine configured to convert data of the first data type of the associated set to the second data type of the associated set, each conversion routines including one or more computer instructions to be executed during conversion;

~~(c) — determining the size of the one or more computer instructions;~~

~~(d) — determining whether the one or more data conversion routines should be generated as stand-alone routines or code chunks based on the size determination; and~~

~~(e)(c)~~ storing said plurality of each data conversion routines in memory accessible to said application program.

9. (Currently Amended) The method of claim 8, ~~wherein said data conversion routines are callable by said application program~~ further comprising determining the size of the data conversion routine for each of the plurality of sets of input attributes and output attributes.

10. (Currently Amended) The method of claim 8, ~~wherein said data conversion routines are stored inline in said application program~~ further comprising determining whether the data conversion routine for each of the plurality of sets of input attributes and output attributes should be callable by said application program or should be stored inline with said application program.

11. (Previously Presented) The method of claim 8, wherein step (b) is performed dynamically while said application program is executing.

12. (Original) The method of claim 8, wherein said input and output attributes are character type.

13. (Original) The method of claim 8, wherein said input and output attributes are date type.

14. (Currently Amended) The method of claim 8, further comprising generating program debugging instrumentation for said ~~plurality of~~ each data conversion routines.

15. (Currently Amended) A system for dynamically generating computer data conversion routines, said system comprising:

a processor; and

a memory device coupled to said processor;

wherein said system is adapted to receive a plurality of sets of input attributes and output attributes from an application program, each set comprising one or more input attributes and one or more output attributes, the input attributes in a set being associated with a first data type and the output attributes in the set being associated with a second data type different than the first data type; and

wherein said memory device stores instructions that, when executed by said processor, cause said processor to:

dynamically create at runtime a ~~plurality of data type~~ data conversion routines for each set of input attributes and output attributes, each conversion routine configured to convert data of the first data type of the associated set to the second data type of the associated set, ~~the each~~ conversion routines including one or more computer instructions to be executed during conversion;

~~———validate specific field conversion options of the conversion routines;~~

~~———determine the size of the one or more computer instructions;~~

~~———determine whether the conversion routines should be generated as stand-alone routines or code chunks based on the size determination; and~~

~~store said plurality of each~~ data conversion routines in a ~~second~~ memory device accessible to said application program, ~~the data conversion routines for converting data from one data type having an input attribute to another data type having an output attribute.~~

16. (Canceled)

17. (Canceled)

18. (Currently Amended) The system of claim 15, wherein ~~said plurality of each~~ data conversion routines ~~are generated~~ is created while said application program is executing.

19. (Original) The system of claim 15, wherein said input attributes are character type and said output attributes are date type.

20. (Currently amended) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor further to generate program debugging instrumentation for ~~said plurality of~~ each data conversion routines.

21. (New) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor to determine the size of the data conversion routine for each of the plurality of sets of input attributes and output attributes.

22. (New) The system of claim 15, wherein said memory device stores further instructions that, when executed by said processor, cause said processor to determine whether the data conversion routine for each of the plurality of sets of input attributes and output attributes should be callable by said application program or should be stored inline with said application program.

23. (New) Logic encoded in a computer-readable medium, the logic operable when executed by a computer to:

(a) receive from an application program at least a first attribute associated with a first input data type and a second attribute associated with a first output data type;

(b) while the application program is executing, dynamically create a first conversion routine based on the first attribute and the second attribute, the first conversion routine configured to convert data of the first input data type to the first output data type, the first conversion routine including one or more computer instructions to be executed during conversion; and

(c) store the first data conversion routine in a memory device accessible to the application program.

24. (New) The logic of claim 23, wherein the first data conversion routine is stored such that the application program calls the first data conversion routine when executing the first data conversion routine.

25. (New) The logic of claim 23, wherein the first data conversion routine is stored inline with the application program.

26. (New) The logic of claim 23, further operable when executed to generate program debugging instrumentation for the first data conversion routine.